Project 1 Readme Team Morales

Version 1 9/11/24

A single copy of this template should be filled out and submitted with each project submission, regardless of the number of students on the team. It should have the name readme_”teamname”

Also change the title of this template to “Project x Readme Team xxx”

<table>
<tr><td>1</td><td>Team Name: Morales</td></tr>
<tr><td>2</td><td>Team members names and netids: Nicolas Morales (nmorale2)</td></tr>
<tr><td>3</td><td>Overall project attempted, with sub-projects: 2-SAT Solver</td></tr>
<tr><td>4</td><td>Overall success of the project: Great! Saw that all cases were within a few milliseconds of each other, regardless of input length, meaning polynomial time was achieved.</td></tr>
<tr><td>5</td><td>Approximately total time (in hours) to complete: 3 hours</td></tr>
<tr><td>6</td><td>Link to github repository: https://github.com/nmorale2/Project1_Morales</td></tr>
<tr><td>7</td><td>List of included files (if you have many files of a certain type, such as test files of different sizes, list just the folder): (Add more rows as necessary). Add more rows as necessary.
<table>
<tr><td>File/folder Name</td><td>File Contents and Use</td></tr>
<tr><td colspan="2">Code Files</td></tr>
<tr><td>2SATSolver_Morales.py</td><td>Main code file. Implements a 2-SAT solver using Tarjan’s algorithm to find strongly connected components (SCCs) and checks for satisfiability.</td></tr>
<tr><td colspan="2">Test Files</td></tr>
<tr><td>2SAT.cnf.csv
kSAT.cnf.csv</td><td>Bunch of test cases. Obtained from Project 1 Files in Canvas</td></tr>
<tr><td colspan="2">Output Files</td></tr>
<tr><td>output_Morales.txt</td><td>Contains the outputs of the program when tested with 2SAT.cnf.csv</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td><table><tr><th colspan="2">Plots (as needed)</th></tr><tr><td>plot_results_Morales.py
execution_time_plot.png</td><td>Gives the time it took to solve each block in the input file.
Resulting plot</td></tr></table></td></tr>
<tr><td>8</td><td>Programming languages used, and associated libraries: Python, csv package, collections package, sys package, subprocess package, time package, matplotlib.plt package</td></tr>
<tr><td>9</td><td>Key data structures (for each sub-project): Directed graph in order to use Tarjan’s algorithm</td></tr>
<tr><td>10</td><td>General operation of code (for each subproject): Finds SCCs using Tarjan’s algorithm, adds implications based on the clauses, then solves the 2-SAT problem using the implication graph and Tarjan’s algorithm. Provides a boolean assignment for satisfiable problems.</td></tr>
<tr><td>11</td><td>What test cases you used/added, why you used them, what did they tell you about the correctness of your code.
I used the 2SAT.cnf.csv file because it was a) readily available, and b) it gave lots of cases. I checked by hand a few test cases to make sure my code was working properly, comparing the two results.</td></tr>
<tr><td>12</td><td>How you managed the code development: Looked online for how to implement a 2SAT solver, then implemented Tarjan’s algorithm, and accompanying functions. Hardest part was putting everything together in order to make the whole thing work.</td></tr>
<tr><td>13</td><td>Detailed discussion of results: Shown below is an example output of two blocks, followed by their inputs.
Time taken for block 0: 0.0554 seconds
Time taken for block 1: 0.0539 seconds

-4,4,0,
1,2,0,

As expected, the length of time is really short, meaning that the SAT solver is really efficient.</td></tr>
<tr><td>14</td><td>How team was organized: I worked alone</td></tr>
<tr><td>15</td><td>What you might do differently if you did the project again: I would try to use a different method of solving the 2SAT instead of Tarjan’s algorithm.</td></tr>
<tr><td>16</td><td>Any additional material: N/A</td></tr>
</table>